HIRLAM 2007 plan for dynamics

Mariano Hortal, INM

The HIRLAM activities on dynamics, as stated in the HIRLAM 2007 work plan, contain the following elements:

- Contribution of HIRLAM to the HARMONIE model dynamics:
  - Investigation of the behaviour of the pressure-gradient term near orography, after introduction of an orographic smoothing similar to the one used at ECMWF.
  - Introduction, subject to the availability of manpower, of a vertical discretization based on finite elements, using geopotential and vertical velocity as non-hydrostatic variables.
  - Investigation of the stability of the semi-implicit method on large areas, using a non-constant map-factor in the linearization, which can be represented with just two or three Fourier components.
- Interfacing the HARMONIE model to the hydrostatic HIRLAM or to the IFS.
  - Different ways of interpolating hydrostatic forecasts to the resolution of 2.5 km (at which the HARMONIE system will be run) will be implemented, tested and compared. At the present, two interpolation tools exist: the HIRLAM tool “gl” and the ALADIN “full-pos”.
  - Adapt the HARMONIE model to use GRIB-2 as its I/O interface.
  - Consider the adaptation of the ECMWF research tool “prepIFS” and of (mini)SMS for preparation, submission and scheduling of HARMONIE experiments.
- The use of transparent boundary conditions both based on the method of characteristics or using a finite-volume approach will be tested in the spectral HARMONIE model.
- A semi-elastic dynamical core (developed by the Estonian group) will be compared in real case forecasts with the fully compressible (ALADIN non-hydrostatic) core of the forecast model.

In addition, some remarks on coding standards. The coding standards we should adhere to have been described by Ryad el-Khatib in a document which has been posted on the HIRLAM wiki pages:
https://hirlam.org/HX/organisation/documents/code_norms.pdf
For me, the most useful rules to make the code more readable are the Doktor naming conventions to identify clearly the nature of each variable (integer, real or character. Local, subroutine parameter or module variable) and to employ USE always with ONLY. In order to increase the readability and to follow the code stream, we should never use GOTO and always indent IF and DO loops.